

STEREO *IMPACT*

IDPU Users Manual

IDPU_UsersManual_C.doc
Version C – 2006-Oct-16

David Curtis, UCB IMPACT Project Manager

Document Revision Record

Rev.	Date	Description of Change	Approved By
A	2002-Sep-18	Preliminary Draft	-
B	2005-Apr-01	More information added	
C	2006-Oct-16	Update information, Add operational info, Add GSE users manual	

Distribution List

Dave Curtis, UCB
 Peter Schroeder, UCB
 Lil Richenthal, GSFC
 Lynn Kistler, UNH

Table of Contents

Document Revision Record	i
Distribution List	i
1. Overview	1
1.1. <i>Introduction</i>	<i>1</i>
1.2. <i>Document Conventions</i>	<i>1</i>
1.3. <i>Applicable Documents</i>	<i>1</i>
2. IMPACT Overview	2
3. Ground System Overview	3
4. IDPU Overview	3
5. IDPU Commands	5
6. IDPU Telemetry	7
7. IDPU Startup	8
8. Memory Map	8
9. Constraints	8
9.1. <i>General Constraints</i>	<i>8</i>
9.1.1. Telemetry bandwidth.....	8
9.1.2. EEPROM Write.....	9
9.1.3. Code Page Select.....	9
9.1.4. Execute Commands.....	9
9.1.5. PLASTIC System Commands.....	9
9.1.6. PLASTIC Software Reset.....	9
9.1.7. Command Sequences.....	10
9.2. <i>MAG Constraints</i>	<i>10</i>
9.3. <i>SWEA Constraints</i>	<i>10</i>
9.3.1. SWEA cover actuator.....	10
9.3.2. SWEA High Voltage.....	11
9.3.3. SWEA Purge (I&T).....	11
9.3.4. SWEA External Cover (I&T).....	11
9.4. <i>STE Constraints</i>	<i>12</i>
9.4.1. STE Bias Supplies.....	12
9.4.2. STE Doors.....	12
9.5. <i>SEP Constraints</i>	<i>12</i>
9.6. <i>PLASTIC Constraints</i>	<i>12</i>
10. Software Maintenance	13
10.1. <i>Building Code</i>	<i>13</i>
10.2. <i>DeltaHex</i>	<i>13</i>
10.3. <i>Hex2Cmd</i>	<i>14</i>
10.4. <i>Loading Code</i>	<i>14</i>
10.5. <i>MemCheck</i>	<i>14</i>
11. IMPACT Operations	15
11.1. <i>Overview</i>	<i>15</i>
11.2. <i>IMPACT Ground System</i>	<i>15</i>
11.3. <i>Operations Personnel</i>	<i>16</i>
11.4. <i>Contingency Planning</i>	<i>16</i>
12. STGSECT USER MANUAL	18

12.1. **INTRODUCTION**..... 18
12.2. **CONFIGURATION FILE**..... 19
12.3. **COMMANDS** 21
12.4. **COMMAND ENCODING**..... 23
12.5. **GSE CONTROLS**..... 24
12.6. **TELEMETRY**..... 24

1. Overview

1.1. *Introduction*

IMPACT consists of a number of instruments connected to the spacecraft via the Instrument Data Processing Unit (IDPU). The IDPU directly controls the Boom suite instruments (MAG, SWEA, STE), and PLASTIC. In addition it serves as the spacecraft interface for the SEP instrument suite. This document shall describe how to run the IDPU and the boom suite instruments. SEP and PLASTIC will have separate Users Manuals describing how to run those subsystems.

1.2. *Document Conventions*

In this document, **TBD** (To Be Determined) means that no data currently exists. A value followed by **TBR** (To Be Resolved) means that this value is preliminary. In either case, the value is typically followed by a code such as UCB indicating who is responsible for providing the data, and a unique reference number.

1.3. *Applicable Documents*

The following documents are closely interrelated with this specification. All documents can be found on the Berkeley STEREO/IMPACT FTP site:

<http://sprg.ssl.berkeley.edu/impact/dwc/>

1. Specifications/IDPUSoftwareRequirements
2. Specifications/PLASTIC_Software_rec – PLASTIC instrument software requirements
3. Specifications/IDPUSpec (IDPU Data Controller Board Specification)
4. ICD/IMPACT_ICD_init_rev (IMPACT/Spacecraft ICD, on the APL web page)
5. ICD/Impact Serial Interface (Instrument Serial Interface Specification)
6. Specifications/DBC04 - IDPU DCB Description
7. ICD/MOC-POC_ICD_Rev- (MOC to POCC ICD)
8. Specifications/SIF06 (SWEA/STE interface specification)
9. Plans/IDPU_SDP (IDPU Software Development Plan)
10. ICD/IMPACT_CTM (IMPACT Command & Telemetry database)
11. TBD (POCC Users Manual)
12. Specifications/CmdFmt_B – IMPACT command formats

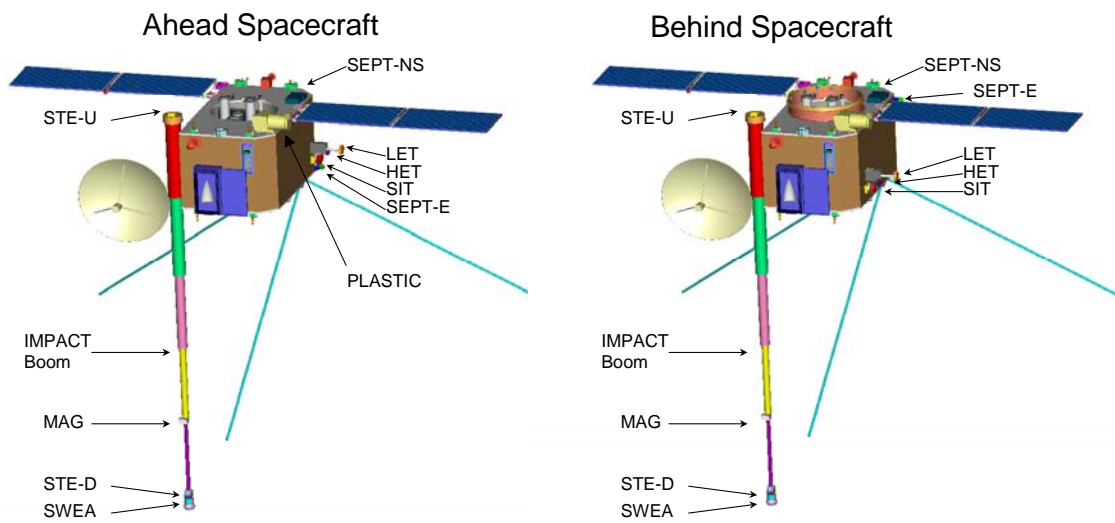
2. IMPACT Overview

The IMPACT (In-Situ Measurements of Particles and CME (Coronal Mass Ejection) Transients) investigation provides the solar wind electrons, interplanetary magnetic fields, and solar energetic particles (SEP) measurements for the STEREO mission. These measurements characterize, with the UNH PLASTIC solar wind ion measurements, the conditions in interplanetary space resulting from a CME observed remotely by the SECCHI imagers and SWAVES radio tracker. The information obtained allows both multipoint measurement-based reconstruction of the interplanetary transient launched by the CME (called the Interplanetary CME or ICME), and as the spacecraft separation grows, the in-situ detection of ICMEs on one spacecraft of CMEs observed leaving the Sun by the imagers on the other spacecraft.

IMPACT consists of seven instruments: SWEA (Solar Wind Electron Analyzer), STE (Suprathermal Electron Telescope), MAG (Magnetometer), SEPT (Solar Electron Proton Telescope comprised of SEPT-E and SEPT-NS), SIT (Suprathermal Ion Telescope), LET (Low Energy Telescope), and HET (High Energy Telescope). The first three of these are located on the IMPACT boom/mast that extends a total of 4.5m antisunward on each spacecraft. (MAG is 3m from the spacecraft, SWEA is at the end of the boom, at 4.5m). The latter four instruments make up the SEP subsystem that is mounted on the spacecraft body. The SEP instruments are mounted at various locations on the spacecraft body consistent with their Field-of-View requirements, as shown in figure 2-1

All instruments use a common IMPACT IDPU that also serves the UNH PLASTIC investigation for data processing and as the spacecraft interface. The IDPU is mounted inside the spacecraft. The space weather (Beacon) data products for all IMPACT instruments and UNH PLASTIC are also accessed through the common IMPACT IDPU spacecraft interface. Initial processing of the SEP data, including the space weather data, occurs in a dedicated SEP package DPU prior to transfer to the IMPACT IDPU.

Figure 2-1 IMPACT locations on STEREO



The IMPACT primary data products are summarized in the Table 2.1.

Table 2-1 IMPACT Science Summary

Experiment	Instrument	Measurement	Energy or Mag. field range	Time Res.	Beacon Time Res. (*)	Instrument provider
SW	STE	Electron flux and anisotropy	2-100 keV	10s	3D x 5E, 60s	UCB (Lin)
	SWEA	3D electron distrib., core & halo density, temp. & anisotropy	~0-3 keV	3D=30s 2D=10s Mom=2s	Moments, 60s; PAD: 2E*12D	CESR (Sauvaud) + UCB (Lin)
MAG	MAG	Vector field	±500nT, ±65536 nT	1/8 s	10s	GSFC (Acuna)
SEP	SIT	He to Fe ions	0.03-2 MeV/nuc	30 s	3S x 4E, 60s	U. of Md. (Mason) + MPAE (Korth) + GSFC (von Rosenvinge)
		³ He	0.15-0.25 MeV/nuc	30 s	----	
	SEPT	Electrons	20-400 keV	60 s	4Dx4E, 60s	U. of Kiel (Mueller-Mellin) + ESTEC (Sanderson)
		Protons	50-7000 keV	60 s	4Dx4E, 60s	
	LET	Ion mass 2-28	1.5-40 MeV/nuc	60 s.	3Sx3E, 60s	Caltech (Mewaldt) + GSFC (von Rosenvinge) + JPL (Wiedenbeck)
		³ He ions flux	1.5-1.6 MeV/nuc	60 s.	2E, 60s	
		H ions flux	1.5-3.5 MeV	60 s.	2Ex2D, 60s	
	HET	Electrons flux	1-8 MeV	60 s.	1E, 60s	GSFC (von Rosenvinge) + Caltech (Mewaldt) + JPL (Wiedenbeck)
		H	13-100 MeV	60 s.	3E, 60s	
		He	13-100 MeV	60 s.	3E, 60s	
Ion mass 2-28		15-60 MeV/nuc	60 s	2S, 2E, 60s		
SEP Common	----	----	----	----	Caltech (Mewaldt) + GSFC (von Rosenvinge)	
IMPACT Common	IDPU (+Mag Analog)	----	----	----	UCB (Curtis)	

3. Ground System Overview

The STEREO spacecraft are controlled by the Mission Operations Center (MOC) at APL in Maryland. The IMPACT suite is controlled by the IMPACT Payload Operations Control Center (POCC) at Berkeley. PLASTIC is controlled by the PLASTIC POCC at the University of New Hampshire. The POCCs communicate with the MOC over the internet using Secure Shell (SSH) and other security measures. The POCC to MOC interface is described in reference 7.

The operation of the POCC is described in reference 11.

4. IDPU Overview

The IMPACT IDPU provides the interface between the STEREO Spacecraft C&DH system and the IMPACT and PLASTIC Instruments. The IDPU software runs on a microprocessor in the IDPU, on the Data Controller Board (DCB). All information transfer between the IMPACT/PLASTIC instruments and the Spacecraft/Ground,

including telemetry, commands, and status flow through the IDPU. The IDPU performs instrument control functions, telemetry compression and formatting, and system monitoring/safing functions.

Figure 4-1 is a block diagram of the IMPACT suite and the IDPU. The instruments communicate with the IDPU over dedicated serial interfaces, as described in reference 5. The instruments are designed to be relatively autonomous. Once their look-up tables have been loaded and their modes and parameters set, they will cycle through their data collection automatically, and provide data to the IDPU without handshaking or other interaction. The IDPU processes the data sent into CCSDS telemetry packets that are sent on to the Spacecraft.

The spacecraft interface is a 1553 bus, as described in reference 4. All telecommands and telemetry are communicated over this interface as CCSDS packets. In addition, status and timing information is provided over this interface.

The spacecraft provides three operational switched and current-monitored power services to IMPACT (IDPU/Mag/STE-U, SWEA/STE-D, and SEP), plus one to PLASTIC. These are controlled by the spacecraft and the Mission Operations Center, and cannot be controlled directly by the POCC. However, the IDPU can request that one or more of these systems be powered off in response to an error condition, and the POCC can trigger such a request via command to the IDPU.

Typically when an instrument is powered off, its survival heater is powered on by the spacecraft. SEP, SWEA/STE-D and PLASTIC have survival heaters. The MAG sensor has a continuous heater that is powered on at all times. SWEA also has an operational heater that is powered off the SWEA/STE-D power service, under control of IDPU software. SEP has a number of operational heaters powered off the SEP power service and controlled by SEP software. Another heater circuit is provided to the IMPACT boom. The heater is controlled by the MOC, and is only on for ~30 minutes prior to deployment to ensure that the boom is warm enough for deployment.

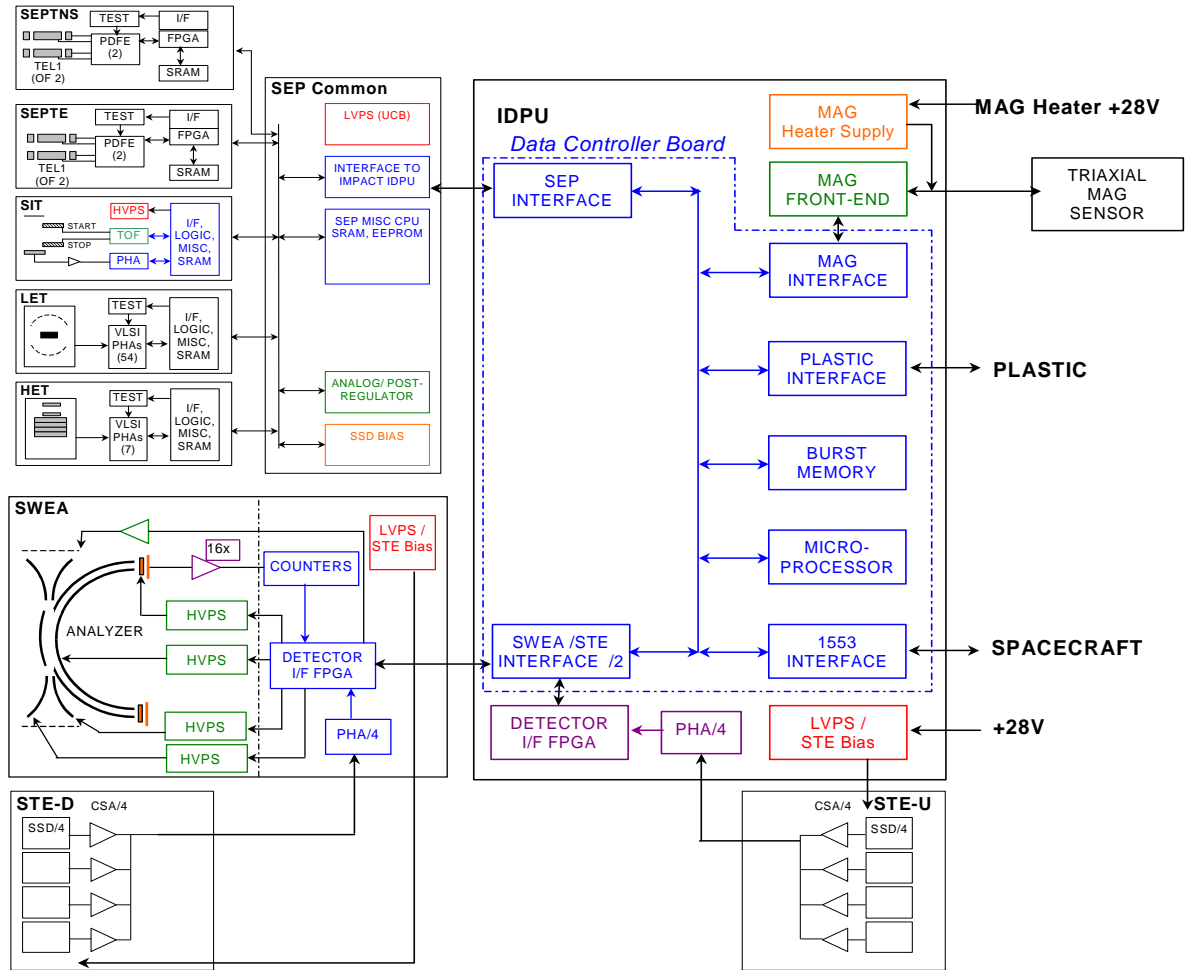
The spacecraft controls most of the IMPACT actuators directly, including the IMPACT boom deployment, SIT and SEPT one-time cover actuators. The IDPU and POCC cannot control these actuators; it must be done by the MOC. The IDPU controls the SWEA (1-time) and STE (reclosable) door actuators. These can be operated by command from the POCC. The STE door shall also be actuated automatically prior to thruster firings and power-down based on status information provided to the IDPU by the spacecraft.

The spacecraft also has 9 temperature sensors that it monitors in the IMPACT suite. These sensors are the only telemetry we get when the IDPU is off. The data from these sensors is part of the spacecraft telemetry, not the IMPACT telemetry, but can be monitored by the POCC. Likewise the current monitors on the spacecraft power services is Spacecraft telemetry.

The IDPU does relatively little for the SEP suite of instruments. Those instruments have their own processing elements, so the IDPU is mostly a bent pipe for their command and telemetry. SEP will have its own Users manual.

While the IDPU runs the PLASTIC instrument, its shall be operated by the PLASTIC team relatively independently of the IMPACT suite. PLASTIC will have its own Users manual describing the details of instrument operation and constraints.

Figure 4-1 Block Diagram of the IMPACT Suite



5. IDPU Commands

The IDPU receives commands formatted in CCSDS telecommand packets. Commands within the IMPACT or PLASTIC ApID (Application Identifier, a code in the CCSDS packet header) range are forwarded to the IMPACT IDPU (see reference 4 and 7). These commands are formulated by the IMPACT and PLASTIC POCCs, sent on to the MOC, then on to the spacecraft, and finally to the IDPU. Commands can be sent real-time or time-tagged. Two kinds of time-tag are accommodated: one interpreted by the MOC,

indicating the earliest and latest time the command can be sent (to accommodate the fact that we are not always in contact), and one interpreted by the spacecraft using a time-tagged command buffer (to allow commands to be accurately timed independent of the contact schedule). The POCC can set either or both of these time tags on the commands it generates.

The IMPACT and PLASTIC range of telecommand ApIDs is further broken down by the task in the IDPU to which it is addressed. Telecommands with SEP ApIDs are forwarded directly to SEP without further interpretation. The IMPACT POCC is constrained to only be able to make send IMPACT ApID commands by the MOC, while the PLASTIC POCC can only send commands with PLASTIC ApIDs. The IDPU is designed such that commands to PLASTIC should not affect IMPACT and vice-versa.

There are a range of System commands, such as memory load, which are normally IMPACT commands. These commands may also be sent with PLASTIC ApIDs by the PLASTIC POCC if need be, but this must first be enabled in the IDPU (by IMPACT command). It is expected that this will not be needed normally.

The IMPACT basic command format and encoding is described in reference 12. The details of the command formats are described in the database in reference 10. The POCC reads the database in reference 10 to encode commands from mnemonics. The database also indicates hazardous commands that require verification before they are sent. An operator should get authorization before proceeding with a hazardous command.

Most IDPU commands are parameter setting commands or function commands. Parameter setting commands set one or more parameters in the parameter table for the task indicated by the command ApID. Parameters include such things as instrument threshold settings and packet generation rates. Function commands perform some immediate function, such as dumping memory or activating an actuator. Details of these tables and functions are indicated in the database, reference 10.

Most commands are fixed length, but some, such as memory load commands, are variable length. The POCC determines the length of the command from the number of parameters entered. The POCC does not know the length of the commands or the size of parameters in the command (byte or word), so care must be taken in formulating commands. The POCC will generate a byte parameter unless it is entered in such a way to indicate it is a word parameter (e.g. 0x0000). See reference 10 for more details. Parameter load commands are variable length, to allow more than one parameter to be modified using a single command. This can cause unexpected results if care is not taken.

Commands can also be provided to the POCC in the form of command messages as described in reference 7. Pre-coded commands in this format can be referenced by file name or can be sent real-time (during I&T or commissioning) via a socket interface in the POCC. This scheme is the method that will be used by the SEP team to provide pre-encoded commands.

6. IDPU Telemetry

The IDPU formats data from the IMPACT and PLASTIC instruments and housekeeping into CCSDS packets as described in reference 7. This data is passed to the spacecraft as described in reference 4, and makes its way back to the POCC from the MOC as described in reference 7.

The IDPU generates packets with ApIDs based on the telemetry source task, with IMPACT data having IMPACT ApIDs, and PLASTIC data having PLASTIC ApIDs. Note also that Beacon data has a specific range of ApIDs which are a subset of the IMPACT and PLASTIC ApID ranges. The spacecraft sends beacon data in real time to any ground station listening, and also records and plays it back (twice) as part of the normal telemetry stream. Non-beacon data may also be sent real-time during DSN passes as well as being recorded and played back (sometimes more than once). All this data will eventually be merged into a single data set for IMPACT.

Packets from the various sources are merged into a telemetry queue and sent on to the spacecraft. If any one data source generates more data than it has been allocated, the queue will back up and telemetry shall be lost. Error messages will indicate this situation, which typically is the result of incorrect parameter settings in the IDPU.

The detailed telemetry formats for the housekeeping and SWEA, STE, and MAG packets are described in the database in reference 10. Each packet has an ApID that indicates where the packet came from and what format the data is in. For some packets other data in the packet must be used to interpret the data, such as sample rate or gain. Some telemetry sources have programmable packet generation rates. This allows some flexibility, particularly during I&T and Commissioning to trade telemetry bandwidth or make use of extra bandwidth provided by the spacecraft.

MAG, SWEA, and STE also make Burst telemetry. This is high time resolution data collected over short intervals (minutes) when the data indicates something interesting has happened. This Burst data is then slowly played back as part of the normal IMPACT telemetry over an interval of several hours. The Burst system is designed to always have something to send, and its telemetry allocation can be finely adjusted to use whatever telemetry bandwidth is available.

The POCC GSE shall display and limit-check the housekeeping telemetry. A separate science display will be capable of displaying instrument data in an engineering format suitable for verifying instrument operation. Data shall be passed on to data analysis systems for generation of higher level products and analysis by scientists. The POCC may be set to get data real-time, or played back over a specific range of ground receive times. Data Analysis processing will be based on merged products downloaded from the MOC or Stereo Science Center (SCC) by FTP.

7. IDPU Startup

In response to a power-up, or a watchdog reset, or a commanded reset, the IDPU starts executing from the boot PROM. The boot PROM contains the command and telemetry handling routines plus the EEPROM writing code, but no instrument code. Only housekeeping and requested memory dump telemetry are generated, and instrument commands will be rejected. The system will normally stay in boot code for 10 seconds, and then copy the first EEPROM code segment to RAM and start executing that. This sequence can be modified during the 10 second window by requesting a different EEPROM segment be loaded (one of four such images in EEPROM), or that code be executed directly out of EEPROM (saves power), or the system can be instructed to continue running the boot code. This allows the operator to avoid running a bad EEPROM code image. The system will not execute an EEPROM code image unless it has a good checksum value.

8. Memory Map

The IDPU contains 8kBytes of PROM, 256kBytes of EEPROM, and 3Mbytes of RAM. I/O and processor registers are also memory mapped. The hardware memory map is described in reference 3 and 6. The processor has only a 64Kbyte memory address space, so the memory is paged – this is also described in reference 3 and 6. Commands for accessing memory must specify a memory page as well as the offset in that page. Typically pages are 16Kbytes in size, and there are 256 such pages in the memory map. However a few commands deal with ‘code pages’, which are 64Kbytes in size.

You can access the I/O registers directly using memory load/dump commands by setting the page to zero. I/O registers are described in reference 6.

Code is typically loaded into the lowest RAM area (page 40h), but boot code gets loaded into 44h. Various parameter tables that control operation have special commands to access them without having to know where they are located in memory. Details of other memory areas are documented in the source code.

9. Constraints

9.1. *General Constraints*

9.1.1. Telemetry bandwidth

Any task with programmable telemetry bandwidth must be organized so that the total IMPACT allocation is not exceeded. Note that on special occasions the IMPACT allocation might be increased. Typically programmable packet generation rates are set by parameter table commands. Command sequences will be set up which alter the telemetry allocations in a consistent manner for various tests. Typically the Burst telemetry rate is adjusted to make the total come out to the available allocation, since the Burst rate can be set to an arbitrary number (/SetBurstRate command).

9.1.2. EEPROM Write

The IDPU software and tables are contained in EEPROM.

In order to write to the EEPROM, there are a number of constraints:

1. The system cannot be running out of EEPROM. Either run from the boot code or copy the EEPROM code to RAM and run from RAM.
2. The EEPROM Write must be enabled before writing, and cleared after writing to the EEPROM (Hazardous command)
3. Special EEPROM Write Commands must be sent, not Memory Write commands.
4. The EEPROM image must conform to the code page standards. This means that the address space from 2000H to 3FFFH must be practically identical to the Boot PROM image.
5. You can patch the EEPROM code as desired, or re-write it all.
6. The EEPROM checksum should be updated prior to trying to run the code. Bad checksums in the EEPROM or RAM code page images are flagged in housekeeping. While writing to EEPROM, all the EEPROM code page checksums in housekeeping will be bad.

9.1.3. Code Page Select

To change from one code page image (EEPROM or RAM) to another after boot can be performed with a Code Page Select command (Hazardous Command). The code page being switched to must conform to the code page standard outlined above. Normally this command is used to switch from an EEPROM page to a copy of that page in RAM. Switching between different code pages is dangerous, and is likely to cause the IDPU to reset. There is also a /RestartToPage command that will jump to the reset vector on a selected code page. /RestartToPage 0 jumps to the PROM page, and is equivalent to a power-on reset. /RestartToPage 0x20 jumps to the reset vector in the first EEPROM image.

9.1.4. Execute Commands

The system allows code at a specified location to be executed once, or executed every polling loop (Hazardous commands). The referenced code must be on the current code page. This system must be used carefully to ensure the code executed is compatible with the system, or the IDPU may reset.

9.1.5. PLASTIC System Commands

As mentioned above, the PLASTIC POCC can be allowed to send System commands (with a PLASTIC ApID) if that ability has first been enabled by the IMPACT POCC. Use the commands /EnablePLASTICSysCmd and /DisablePLASTICSysCmd to control this feature (from the IMPACT POCC).

9.1.6. PLASTIC Software Reset

In the event that the PLASTIC software is not behaving, you can reset just the PLASTIC software without affecting the IMPACT software. This avoids having to reconfigure all of IMPACT in order to fix some problem with PLASTIC. Send the command

/RestartPLASTICSoftware from the IMPACT POCC (or the PLASTIC POCC is PLASTIC System Commands have been enabled).

9.1.7. Command Sequences

The IDPU has 32 command sequences it can execute in response to some event. The sequences contain a list of commands in a slightly compressed format. Any number of sequences can be active at the same time. Generally one command from each active sequence is executed each second. A special command can be embedded in a sequence to cause execution of the sequence to be paused for a selectable number of seconds.

Some command sequences are connected to specific events, such as instrument power-down warning. In addition any sequence can be started by command, including from another sequence (/StartCmdSeq). An active command sequence can be aborted (/AbortCmdSeq). If not aborted the sequence will run till it reaches the last command in the list, and then go inactive. Command sequence will only start if they have been enabled (/EnableCmdSequence, /DisableCmdSequence).

Reference 10 describes the nominal command sequences. The default command sequences are loaded from EEPROM on start-up. They can be modified by a memory load command. The POCC can build such a memory load command from a command script. Command sequences are limited to 128 bytes of commands each.

9.2. **MAG Constraints**

Do not power on the IDPU or MAG Heater without a MAG sensor attached.

Sometimes the MAG sensor comes up in a saturated condition. This can be cleared by sensing a /MAGEnableIF command.

9.3. **SWEA Constraints**

9.3.1. SWEA cover actuator

The SWEA instrument has a 1-time cover opening actuator. Once opened, this cover can only be re-closed manually (not at all after launch). This cover is designed to prevent contamination of the sensitive Micro Channel Plate (MCP) detectors in SWEA. Typically this cover shall be opened during commissioning, no sooner than 24 hours after launch (to allow time for the spacecraft to out-gas). The cover will also be opened during I&T, but only when measures have been taken to ensure the MCP are not contaminated. The SWEA cover opening requires two commands (ARM and Execute); the ARM command is hazardous, and should not be sent without authorization. The Execute command must follow the Arm command within a few seconds or the hardware will time out, and the Execute command will be ignored.

An enable plug on the side of SWEA disables the SWEA door actuator (this is the same enable plug used to enable SWEA High Voltage and the STE door actuators). Normally, during I&T, this enable plug will be out, providing another level of safety against accidental cover opening. This enable plug must be in for launch, and for any test

requiring the door to be opened. The plug should not be installed without authorization. There are two enable plugs; a red-tag “Test” enable plug and a green-tag “Flight” enable plug. Both plugs enable door actuation.

9.3.2. SWEA High Voltage

The SWEA contains High Voltage (up to 3000V). No high voltage is exposed, so there is no hazard to personnel. However, if the MCP high voltage is applied when SWEA is not in a good vacuum ($<1E-5$ Torr) there is a risk of arcing in the instrument that could damage the detectors and electronics.

There are two high voltage supplies: the MCP supply and the Non Regulated (NR) supply that is the basis of the SWEA analyzer electrostatic optics voltages.

To enable the SWEA High Voltage supplies requires two commands (Arm and Execute); the ARM command is hazardous, and should not be sent without authorization. The Execute command must follow the Arm command within a few seconds or the hardware will time out, and the Execute command will be ignored. Once enabled, the SWEA MCP supply can be programmed to the desired voltage by ground command, and the SWEA analyzer supply cycling starts.

The MCP supply should not be changed without authorization.

An enable plug on the side of SWEA disables the SWEA high voltages (this is the same enable plug that enables the SWEA and STE door actuators). Normally, during I&T, this enable plug will be out, providing another level of safety against accidental application of high voltage. This enable plug must be in for launch, and for any test requiring high voltage. The plug should not be installed without authorization. There are two enable plugs; a red-tag “Test” enable plug and a green-tag “Flight” enable plug. Both plugs enable door actuation. The Test plug enables everything except MCP high voltage; the Flight plug enables everything. Note that running the NRHV system in air is not hazardous to the instrument or personnel.

9.3.3. SWEA Purge (I&T)

During ground operations, the SWEA shall be purged to prevent the build-up of contaminants on the MCP detector. Purge (5L/hour) shall be provided on a nearly continuous basis (outages of up to 24 hours can be accommodated). Purge shall be provided via a separate purge fitting rather than the spacecraft purge manifold due to boom deployment issues. Just prior to encapsulation (pre-launch) purging shall be discontinued and the SWEA/STE-D purge manifold removed.

9.3.4. SWEA External Cover (I&T)

SWEA has an external non-flight cover to protect the delicate external grids and keep the aperture clean. This cover will be on for most of I&T, but must be removed before launch, probably just before encapsulation.

9.4. **STE Constraints**

There are two separate STE instruments; STE-U (upstream, sunward FOV, mounted on the boom at the sunny end), and STE-D (down-stream, anti-sunward FOV, mounted with SWEA at the end of the boom). These two can be operated fairly independently, but their data products are combined.

9.4.1. STE Bias Supplies

Each STE unit has a bias supply for setting the detector bias. This supply is programmable using the command /SetSTEUBias or /SetSTEDBias followed by an 8-bit DAC value, 0-255 corresponding to 0 – 140V. The default DAC value is 26, which is the minimum value to get reasonable resolution out the system. A higher bias may be desirable; this will be determined during commissioning and updated periodically (there is some evidence that a higher bias may be needed to mitigate radiation damage to the detector over time).

9.4.2. STE Doors

Each STE unit has a door mechanism to protect the detector from contamination or direct exposure to sunlight (which could overheat the detector). The door may be opened or closed by command (/STEUDoorOpen, /STEDDoorOpen, /STEUDoorClose, /STEDDoorClose). STE-D door actuation is only possible if either the Test or Flight enable plug is installed into the SWEA pedestal. Door actuation has a number of constraints imposed by the IDPU to prevent damage to the actuator wire. The door cannot be actuated if the STE unit is too warm, the door actuation has a maximum actuation interval, and the door cannot be actuated before the system has had time to cool off from the previous actuation. All these parameters are programmable by table load, but should not be altered without authorization. The default parameters are appropriate for normal on-orbit operation (STE cold, in vacuum). There are other sets of parameters in scripts for other specific situations, most notably for in-air, room temperature testing (script file STEDoorAir.cmd). For ground testing in air you will need to run this script before attempting to actuate the door (if not, an error will be generated).

After normal operations are reached, IDPU Instrument Control Scripts will be loaded to automatically actuate the STE doors in response to thruster firing warnings, off-pointing warnings, and power-down warnings from the spacecraft. The default IDPU Instrument Control Scripts do not open the STE doors, but will close them in response to power-down, thruster operation, or off-pointing if those sequences have been enabled.

9.5. **SEP Constraints**

The IDPU is mostly a bent-pipe for SEP. There are no IDPU constraints for SEP. SEP instrument constraints are documented in the SEP Users manual.

9.6. **PLASTIC Constraints**

PLASTIC constraints are documented in the PLASTIC Users Manual.

10. Software Maintenance

The IMPACT and PLASTIC code is written in Intel 80196 assembly language. The code has a number of constraints that are not obvious. Do not make changes to the code, particularly the system code, without a good understanding of those constraints. Any code modification should first be tried out on an IDPU ETU unit, and it should pass an Acceptance test before being loaded onto a flight unit.

10.1. *Building Code*

Software is developed using the Phyton PDS-96 development tool. Project IMPACT_EEPROM_A is used to generate the code for the Ahead spacecraft and IMPACT_EEPROM_B for the behind spacecraft. The released version of this code resides on computer 'Curtis' at U. C. Berkeley Space Sciences Laboratory. It is in a subdirectory C:\Users\Stereo\Designs\FSW\. The version number should be updated in file Build.inc in subdirectory EEPROM_A (Ahead) or EEPROM_B (Behind) before building the code. PDS-96 will build the following files in a subdirectory of the source directory listed above:

Ahead spacecraft, Project IMPACT_EEPROM_A:

```
EEPROM_A\IMPACT_EEPROM_A.HEL
EEPROM_A\IMPACT_EEPROM_A.HEX
EEPROM_A\IMPACT_EEPROM_A.HEE
EEPROM_A\IMPACT_EEPROM_A.HEH
```

Behind spacecraft, Project IMPACT_EEPROM_B:

```
EEPROM_A\IMPACT_EEPROM_B.HEL
EEPROM_A\IMPACT_EEPROM_B.HEX
EEPROM_A\IMPACT_EEPROM_B.HEE
EEPROM_B\IMPACT_EEPROM_B.HEH
```

These are Intel Hex files, which together define the EEPROM memory image for one code page.

10.2. *DeltaHex*

The next step is to combine the HEX files. This is done by a program called DeltaHex. This program can either build a full image, or build a 'Delta' image containing only the differences between the new code and the most recent version (as defined by binary file DeltaHex.dat, generated whenever a new baseline is built).

The input to the DeltaHex program defaults to the files identified in DeltaHex.cfg, typically one of the two lists of 4 files indicated above. That can be changed before running DeltaHex, or in the course of running DeltaHex using its user interface. When you quit DeltaHex, DeltaHex.cfg is updated.

The output file is called DeltaHexBaseline.hex (for a new baseline) or DeltaHex.hex (for a delta). These files are again Intel Hex files and end up in the DeltaHex directory

(where the DeltaHex program resides), nominally
C:\Users\Stereo\Designs\GSE\DeltaHex.

10.3. **Hex2Cmd**

Hex2Cmd takes an Intel Hex file and converts it into a command script to be loaded into the IDPU by the stgsect POC program. Hex2Cmd starts by asking for the source file; you navigate to the DeltaHex directory and select the source file, typically DeltaHexBaseline.hex.

Hex2Cmd will next ask you what memory page the code is to be loaded into (in hex). Typical values are 40 to load into RAM (from Boot code), 20 for the first EEPROM image, 24 for the second 28 for the third, and 2C for the fourth.

Hex2Cmd then generates a file with the same name but ending in .cmd.

Typically the output file is renamed after running the program (so it is not over-written the next time you run). The normal file name is something like:

Code20050225V25_EEPROM0.cmd

Indicating code generated on February 25 2005, version 25, to be loaded into the first EEPROM page.

10.4. **Loading Code**

Now you can run stgsect and run the command script with the Start command; navigate to the directory containing the .cmd file and select it. It typically takes a few minutes to load. If you are loading an EEPROM image, first send the command '/EEPROMenable'. After loading, you need to update the code checksum using '/UpdateChecksum #', where # is 0 for the first EEPROM code page, 1 for the second, 2 for the third, 3 for the fourth, and 4 for the RAM code page. Finally send '/EEPROMDisable'

10.5. **MemCheck**

The MemCheck can be used to compare a memory load to a memory dump to verify the load. First dump the code page. For example, to dump the first EEPROM image do the command '/DumpMemory 0x20 0x0000 0x0100'. This will take a while – 256 packets at one packet every 10 seconds typically. Note the name of the data file that stgsect is currently logging telemetry to. Then run the MemCheck program. Push the 'ReadLoad' button and select the name of the file containing the memory load (.cmd file). You can verify the range of memory that is loaded on the display. Next press the 'ReadDump' button and select the telemetry file (.tlm) containing the dump. The system then compares what is in the dump with what is in the load and tells you if they match. #loaded should be the same as #verified and #wrong should be zero.

11. IMPACT Operations

11.1. Overview

The IMPACT instrument suite is operated with an enhanced version of the existing IMPACT Command and Telemetry GSE used throughout Observatory I&T, instrument testing and mission simulations. POCC hardware will be located at UC Berkeley's Mission Operations Center with identical backup hardware located at APL. The POCC software is used to create, send and verify commands to the IMPACT suite. The SEP team creates its own command sequences which are forwarded to UC Berkeley for verification and then sent through the POCC software to the spacecraft. A database of contingency responses has been created to guide the POCC operators.

11.2. IMPACT Ground System

IMPACT's POCC software is based on the Command and Telemetry GSE used throughout mission development. The POCC hardware consists of two Windows XP computers (one each for the Ahead and Behind spacecraft) located within the secure area of the Mission Operations Center at UC Berkeley's Space Sciences Laboratory. In addition, identical backup hardware is located at APL for use in the event that UC Berkeley becomes incapable of commanding. While each computer is assigned to either the Ahead or Behind spacecraft, the software running on both is the same providing additional redundancy in the event one of the two commanding computers fails.

The main functions of the POCC software are to generate command sequences, to transfer command sequences to the MOC at APL, and to monitor instrument state of health. In addition, the POCC system is capable of forwarding real-time telemetry to other computers and locations running science display modules. The telemetry forwarding mechanism is crucial as the IMPACT team is large and diverse, and the team members with the most engineering knowledge of certain IMPACT subsystems are not located at UC Berkeley. For security reasons, the computers receiving telemetry through this forwarding process are not capable of directly sending command sequences to the MOC. All sequences must be handled by the operator at the POCC.

The POCC software's commanding functions include the encoding of command mnemonics into Supplemented Command Messages (SCMs) and the option of sending command sequences with time tags. Time tagged commands are uploaded to the spacecraft from the MOC and stored in the spacecraft's onboard stored command buffer until forwarded to the IMPACT IDPU at the execution time set by the time tag. In addition, the software is capable of running command scripts written in a subset of STOL and of forwarding prepackaged SCMs from the SEP team. The POCC verifies receipt of the commands by the IMPACT IDPU using instrument housekeeping telemetry in real time. Finally, the POCC automatically creates a log of all commands sent to the MOC.

The POCC software also monitors state of health telemetry during passes. This telemetry is collected, decoded, displayed and limit-checked by the POCC. In the event that a state of health quantity is outside its limits or an IDPU error is detected, the POCC notifies the operator through an automatic paging system. Operator response to such an event is

guided through the use of a database which catalogs responses to all potential limit violations and errors.

11.3. *Operations Personnel*

During the commissioning phase of the STEREO mission, the Project Manager Dave Curtis is primarily responsible for IMPACT commanding assisted by Peter Schroeder, IMPACT's Operations/Data Manager, and other members of the IMPACT team as necessary. Once the STEREO spacecraft enter their heliocentric orbits, Peter Schroeder becomes primarily responsible for IMPACT commanding assisted by IMPACT team members as required. The POCC software is not capable of creating sequences specific to the SEP suite. Instead, the SEP Central team headed by Caltech is responsible for producing SEP-specific command sequences.

11.4. *Contingency Planning*

As noted above, a database has been compiled of all potential limit violations and instrument errors with corresponding actions agreed to by the IMPACT team. In addition, if the primary course of action is to contact an IMPACT team member, a backup action exists in the event that the team member is not available.

When a contingency situation is detected by the POCC, a text page with specific information regarding the contingency is sent to the operator. If the MOC detects an anomaly related to the instrument, the MOC also has the ability to use this text paging system by sending an email to the appropriate address. If a contingency requires that commands be sent to the spacecraft, the MOC is immediately notified of the situation and the relevant IMPACT personnel are notified. POCC operating personnel are typically capable of responding to a contingency situation within 20 minutes of receipt of the page if the personnel are home rather than at UC Berkeley.

The following contacts are to be used when a contingency occurs:

IMPACT text pager (pages IMPACT POCC operator on duty):
510-442-0286 or 5104420286@myairmail.com

Boom Suite (SWEA, STE, MAG, IDPU):

Dave Curtis, 510-642-5998 (office), 510-918-3047 (cell)
Peter Schroeder, 510-643-9591 (office), 510-828-2933 (cell), 510-338-4139 (home)

SEP Central and LET:

Branislav Kecman, 626-395-4264 (office), 626-354-2534 (cell), 626-797-0470 (home)
Alan Cummings, 626-395-6708 (office), 626-975-2469 (cell), 626-355-2706 (home)

HET:

Tycho von Rosenvinge, 301-286-6721 (office), 443-995-6019 (cell), 410-280-2470 (home)
Kristin Wortman, 301-286-9454 (office), 410-336-9962 (cell), 410-489-3621 (home)

SEPT:

Stephan Boettcher, +49-431 880 2508 (office), +49-160 9136 6074 (cell), +49-4305 99 17 72 (home)

Reinhold Mueller-Mellin, +49-431 880 3227 (office), +49-174 756 6121 (cell), +49-431 37 21 18 (home)

SIT:

Glenn Mason, 240-228-2805 (office), 301-681-6598 (home)

Tycho von Rosenvinge, 301-286-6721 (office), 443-995-6019 (cell), 410-280-2470 (home)

12. STGSECT USER MANUAL

12.1. INTRODUCTION

STGSECT is command and display software (POC) which connects to the MOC via TCP/IP sockets. It displays telemetry in hex form and uses a database to convert the housekeeping telemetry into units and/or text for display with over/under limits checking; housekeeping values can be plotted vs time. For commanding, STGSECT uses a database to convert mnemonics into SCM's, provide hazard warnings, and send them to the MOC. The command line accepts a subset of STOL commands, providing local variables and comparisons for flow control, and control of various software switches. All commands can be executed from script files and script files can be run according to a schedule file. A log is displayed showing time, date, commands, limit violations, errors, and comments; the log is saved to a disk file. A Command Verification system checks that commands sent and verified by the MOC match commands reported by the IDPU.

A telemetry server allows up to 20 other modules to receive telemetry via TCP/IP socket. A command server allows up to 10 other modules to send SCM's for immediate relay to the MOC or storage on the local hard drive to be sent later by remote or local command. Before the SCM's are sent to the MOC, sequence counts are set and new checksums added.

STGSECT can be connected to another STGSECT's command and telemetry servers instead of the MOC servers, to provide a remote control; commands can be sent, scripts run, and telemetry displayed. When STGSECT is connected to another STGSECT, commands sent by the STGSECT that is connected to the MOC will be shown as "Unexpected Commands" in the remote STGSECT. The remote STGSECT will not have the updated command sequence so subsequent commands from the remote STGSECT will accumulate in the FIFO even though they are verified in the other STGSECT. If the command sequence is manually updated, the next command will be verified and the remaining commands will be shown as "dropped commands."

When STGSECT starts, there are three windows visible; The Event Log with a Command Line, the Command window with a list of command mnemonics and a Command Buffer window for scripts in progress, and the Received Data display with a hexadecimal data window. The Event Log Title bar will indicate which Spacecraft is sending telemetry after the first packet is received. There are two kinds of commands: Commands to control the program and commands that are sent to the MOC to be forwarded to the instruments. All commands can be typed into the command line or run from a script file.

12.2. CONFIGURATION FILE

A number of choices must be made by means of a configuration file in ...\`stgsect\cfg`. This file should be named "`stgsect.cfg`". The line order must be maintained. The part of each line before the "=" may be changed as desired to describe the purpose of the line. Here is a line by line description of the contents of the configuration file (the descriptions in parenthesis are not part of the line):

0 Command Server=128.32.147.63	(Name or IP address of the command server, normally the MOC.)
1 Telemetry Port=5501	(Port number for telemetry server. The server name is line 12 below).
2 Command Port=5500	(Port number for command server.)
3 Command CCSDS Header Swap=0	(Command CCSDS Header is swapped if 1, not if 0.)
4 1958 epoc adjust=1830297600	(Adjusts EPOC from 1-1-1900 to 1-1-1958)
5 Telemetry CCSDS Swap=1	(Telemetry CCSDS Header is swapped if 1, not if 0.)
6 Telemetry Forward Port=5001	(Port number to use for registering telemetry forwarding server.)
7 Command Configuration file=cmd0.cfg	(Filename in "..\ <code>stgsect\cfg</code> "; one button command panel neponics.)
8 Seconds to wait for ARR=60	(Timeout waiting for an ARR from the MOC.)
9 SCM receive port=5002	(Port number for SCM receiving server.)
10 SEP Command CCSDS Header Swap=0	(Command CCSDS Header is swapped if 1, not if 0.)
11 IMPACT = 1	(Command Verification uses IMPACT parameters if 1, PLASTIC if 0.)
12 TELEMETRY SERVER = 128.32.147.63	(Name or IP address of the telemetry server, normally the MOC.)
13 SCM Timeout seconds = 60	(Timeout waiting for an SCM to completely arrive.)
14 Initial alarm delay (254=max, 255=off)=5	(Number of telemetry packets received before limit checking starts.)
15 Alarm delay counter (255 max) = 0	(Number of consecutive limit violations before alarm is displayed.)
16 Email server (Apollo) = 128.32.147.81	(Name or IP address of alarm notification email server.)
17 Email sender = sender@ssl.berkeley.edu	(Email address of alarm notification sender.)
18 Email receiver = recipient@ssl.berkeley.edu	(Email address of alarm notification recipient.)
19 EEPROM0 code page = 0x20	(Code Page setting for command sequence uploads.)
20 EEPROM1 code page = 0x24	(Code Page setting for command sequence uploads.)
21 EEPROM2 code page = 0x28	(Code Page setting for command sequence uploads.)
22 EEPROM3 code page = 0x2C	(Code Page setting for command sequence uploads.)
23 RAM code page = 0xE000	(Code Page setting for command sequence uploads.)
24 Command Sequence offset = 0xF000	(Memory offset for command sequence uploads.)
25 Record Telemetry to file = 0	(Default for recording telemetry to disk; 0 is off, 1 is on.)

26 Reconnect attempts = 3

(If telemetry stops, number of
disconnect/reconnect attempts)

27 PLASTIC log, email error off = 1

(Set to 1 to stop all PLASTIC email error
notifications)

12.3. **COMMANDS**

These commands can be typed on the command line or run from script files. The script files are ascii text and are typically named with a .cmd extension and placed in a subdirectory named cmd (...\\Stgsect\\cmd\\somescript.cmd). The following list of commands is available from within the program: click help on the menu.

BADCKS on/off: If on, makes bad checksums for all SCM's (Supplemented Command Message). Defaults to off.

CMD mnemonic (or /mnemonic): Assembles SCM based on database and sends it to the command server (MOC).

ECHO on/off: Turns the Event Log display on or off. Defaults to on.

ENDPROC: Clears all pending script commands.

ENDSEQUENCE: Ends recording of command sequence started by STARTSEQUENCE.

FREE VariableName: Removes user assigned variable name.

GO: Cancels any wait.

IF(PARAMname or SOHname == value or VarName) cmd mnemonic, cmd mnemonic, ...max=10: Also !=, >=, <=, > or <. Integer values: SOHname & intvalue == value or VarName. Also |. SOHname is the label of an item on one of the SOH menu panels.

LET VariableName = value or SOHname: Assigns value to variable.

LOAD filename: Loads 16 button command panel from file.

The files, named cmd0.cfg, cmd1.cfg etc. are in the "cfg" sub directory.

LOG on/off: Turns logging of Event Log to disk file on or off. (.log)

PAGE filename: Recalls previously saved screen configuration from file. (.srd)

QUIT: Ends program.

REC on/off: Turns on/off recording of received telemetry to disk.

REM text: "Text" is a comment that will be echoed and logged if echo and log are enabled. ; text "Text" is a comment that won't be echoed or logged.

SAVECMD on/off: Starts and stops command recording to disk file (.svc).

SCHEDULE filename: Loads file with schedule of commands and script files to run at listed times.

SENDSCM on/off: Allow or not allow sending SCM's while a script is running.

SEND filename: Sends SCM file containing one SCM to the command server.

SET ParameterName value: Sets various parameters. See SET Parameters below.

START filename: Starts script. Script commands can be nested.

STARTSEQUENCE codepage sequence filename: Starts Command Sequence mode; subsequent commands are sent to a command sequence file instead of the instrument. Code Page: EEPROM0 to EEPROM3 or RAM. Sequence Number: 0 to 32.

WAIT: No parameter means indefinite wait.

WAIT n: Wait n seconds, then continue.

WAIT UNTIL SOHname = value or VarName: Waits until condition is true. Also !=, >=, <=, > or <. Integer values: SOHname & intvalue = value or VarName. Also |.

SET PARAMETERS (These can be set from the command line or from a script.)

SET MOC ascii directive: Sends directives to the MOC or EMULATOR. (This command attempts to connect to the server if not already connected.)

SET MOC OFF (Disconnects from the Telemetry Server.)

SET TSRV TelemetryServerName/IP address, port number

SET SCM scmname scmvalue:

msgt message_type (e.g. SCM-)

cmdm command_message_format_version (e.g. 00)

spar spare (e.g. 000)

faci facility_ID (e.g. SWAVES , IMPACT , PLASTIC , SECCHI)

spac spacecraft_ID (e.g. EA for ahead, EB for behind)

pack decimal_number_of_packets (e.g. 01 to 99)

enab enable_time_GMT_Time (YYYYDOYHHMMSS all zeros if real time, dashes if not used)

time time-out_GMT_Time (YYYYDOYHHMMSS , dashes if not used)

delt delta_time-out_seconds_after_enable_when_MOC_stops_sending_cmd (00000 to 86400)

SET CCSDS ccsdsname ccsdsvalue:

vers version number 3 bits, 000

type packet type 1 bit, 1 for telecommand packet

seco secondary header 1 bit, flag 1 for header, 0 for no header

flag sequence flags 2 bits, 11 for standalone packet

coun sequence count 14 bits, 0 to 16,383, no enable time; no secondary header

ecou sequence count 14 bits, 0 to 16,383, enable time; no secondary header

rcou sequence count 14 bits, 0 to 16,383, no enable time; secondary header

ercou sequence count 14 bits, 0 to 16,383, enable time; secondary header

secs command routing time 32 bits, seconds since epoc, all 0 means forward immediately

subs command routing time 8 bits, sub seconds

12.4. **COMMAND ENCODING**

When it is started, the GSE loads the command databases into memory. These databases are text files that contain a mnemonic followed by values separated by spaces which make up the command. The values are the APID followed by command data. These are the required files:

BASIN.TXT	Spacecraft telemetry display name information.
BASIS.TXT	Spacecraft telemetry definitions
impact_apd.txt	Impact APID definitions
impact_ctm.prn	Impact command definitions (Includes PLASTIC commands)
impact_err.txt	Impact error definitions
impact_seq.txt	Impact command sequence definitions
impact_tbl.txt	Impact table definitions
impact_tlm.txt	Impact telemetry definitions
PLASTIC_dhk.txt	PLASTIC digital housekeeping definitions
SEP_HKG.txt	SEP housekeeping definitions

A line beginning with REM is a comment that will appear in the log window and in the log file. A line that begins with ; is a comment that is ignored. Commands begin with / and a mnemonic. Commands can be sent by typing them on the GSE command line or by selecting them from the GSE command panel. The command data may be entered as hex (preceded by 0x), decimal, and ASCII text (enclosed in quotes). Clicking an item in the command window and pressing ENTER puts an entry in the command line where it can be modified and/or sent. Commands can also be entered directly without using the mnemonic database: e.g. /0x220 0x1234 0001 "AB" -1.

Instrument commands can also be sent from script files. The lines of the script start with / followed by a command mnemonic and parameter values. There is a START button on the event log panel just above the command line which can be used to select and run a script stored on the hard disk.

Each command is encoded into a supplemented command message (SCM) and sent to the command server. The SCM header has some values that are fixed and some values that are set automatically. Other values may be changed by script commands or by changing them on the SCM panel (SCM on the menu). In scripts, the **SET** command is used to change the values. (e.g. SET SCM VER 002; this sets the version to 2).

Pre-formed SCM's from another source (as in the case of SEP SCM's) can be sent to the GSE via the SCM server port. The GSE will check to see whether the SCM should be sent immediately or saved to disk for sending in the future. When an SCM is to be sent, its sequence number is updated and a new check sum is added before sending the SCM to the command server. When receiving a long SCM via the SCM server port, the received data may be in more than one piece; there is a time out setting in the configuration file to determine how long the GSE will wait for the SCM before logging an error. Pre-formed SCM's that have been saved on the disk can be sent by using the SEND command; this command can be used via the SCM server.

When an SCM is sent to the command server, the GSE waits for an ARR (Authorization Return Receipt) from the command server before sending another SCM. The amount of time the GSE waits for the ARR before logging an error can be set in the configuration file.

All commands are logged in the event log and the event log is saved in a disk file. The complete SCM's can be saved in hex form in a separate disk file by toggling on command saving with the command: **SAVECMD** on/off.

12.5. **GSE CONTROLS**

CANCEL button ends a script. This button can be used to cancel a script while it is running.

EM toggle button turns on/off email limit alerts; right click to temporarily change parameters. Parameters can be changed permanently by editing lines 16, 17, and 18 in the configuration file.

GO button cancels any wait.

RECORD button on Received Data Panel toggles recording of received data. The file is named: A or B date and time with extension .t1m. (A or B for selected spacecraft; this setting is in line 26 of the configuration file.

START button opens file select dialog with *.cmd (Script files).

WAIT button initiates an indefinite wait; button is dimmed when any wait is in progress.

SWAP button on the Received Data Panel byte swaps the hex display from the 27th to the 298th byte.

State of health panels can be hidden by a right click on the panel.

Up arrow recalls last 10 command lines.

F12 clears command line buffer.

ESC clears the command line.

Printing: The print item on the top menu bar prints the main GSE panel and all visible child panels. To print an individual panel, double click in its upper left corner; the print dialog will appear. The default orientation is landscape. The larger dimension of the panel should either be set to fill the paper or set to a specific size in tenths of a millimeter. The other dimension should be set proportional to the first. The long dimension of the paper is the height, the short dimension is the width.

12.6. **TELEMETRY**

The Received Data panel displays received telemetry packets in hex form. The ON/OFF button marked TLM is for connecting to the Telemetry server (MOC). The first time it is clicked, connection is attempted based on the IP address or name and port number in the configuration file. The next click disconnects, and the next click allows reconnection with the default IP address or name and port number or a new user entry. After a connection is made to the Telemetry server, click the ON/OFF button marked Display to view telemetry. The Apid window can be set to show only a particular APID; the default is FFF which means all APID's. There are two more hex display windows available on the "main" menu.